# Novel Rate-Jitter Control Algorithms

Modeling and Analysis

Madhu Babu Sikha, Manivasakan Rathinam
Department of Electrical Engineering,
Indian Institute of Technology Madras,
Chennai-600036, India.
email: ee10d028, rmani@ee.iitm.ac.in

***Abstract*—Time Division Multiplexing over Internet Protocol (TDMoIP) is a transport technology that extends the voice, video and data traffic transparently over IP. When TDM traffic is packetized and injected into a packet switched network (PSN) for transportation, packets arrive at the destination with different inter-arrival times, due to variable delay (jitter) introduced by PSN. This network induced jitter should be minimized, because TDM devices operate at constant bit rate. Problem of jitter from theoretical (competitive and statistical analysis) to more practical view point had been well studied. In this paper, we have proposed two on-line algorithms, algorithm-A and algorithm-B to minimize rate-jitter. We have shown both analytically and by simulation that the rate-jitter achieved by algorithm-A is strictly less than the rate-jitter of an on-line algorithm proposed in a previous work. The simulation results shows that algorithm-B also achieves less rate-jitter than the reference algorithm. We also undertake the statistical analysis of above algorithms, in particular, we have modeled jitter buffer as an $\mathrm{MMPP}/\widetilde{\mathrm{D}}/1/\mathrm{B_{on}}$ queue by making two assumptions on Markov modulated Poisson process (MMPP) and steady state queue length distribution is calculated. The correctness of the analytical results corresponding to our proposed algorithms is verified with simulation results. From the simulation results, it is also shown that the mean waiting time of a packet in the buffer is less for both proposed algorithms compared with the reference algorithm.**

***Keywords-TDMoIP; MMPP; rate-jitter; PSN; state dependent service***

## I. Introduction

TDM [19] has been the most promising technology over the decades to transmit voice. In the recent years, it is being used to transport video and data also. In TDM, there is a dedicated channel for each user. This channel is used only when the user is making a call or when some data is transmitting. Therefore, the bandwidth is not used effectively in TDM, since the channel is idle most of the time and TDM services are also expensive. On the other hand, PSN [20, 21] uses bandwidth efficiently and is cheap. So, emulating the TDM traffic over a PSN is an effective solution.

TDMoIP is a mechanism to connect two TDM islands through IP network. On the transmitter side, fixed number of TDM frames are packetized and sent across an IP network. So, all the packets with TDM payload are of equal size. At the receiver, the TDM payload is retrieved along with timing and the TDM stream is regenerated before sending downstream. If the TDM payload is sent over connectionless service in IP, reordering is done. When these IP packets with TDM payload traverse through the network, each packet may be routed in different path, so, they encounter different nodes and variable queueing delays. This queueing delay is the dominant part in end-to-end delay of a packet. Finally, they arrive at the receiver with variable inter-arrival times (IATs) as compared with almost constant IAT at the transmitter. This variation in the arrivals (packet delay variation) is called as *jitter*. At the receiver, this jitter causes serious problems for audio playback. To overcome jitter, all the received packets are stored in a buffer called as *jitter-buffer* and associated an algorithm which decides the dequeueing instant of next packet to be transmitted (or service initiation time of each packet). The above algorithm minimizes the output jitter, given the arrival times and/or the number of packets in the queue. This process is called as jitter regulation. The output of jitter regulator is fed to a link scheduler to send the packets on to an outgoing link. Jitter control is the sequence of the two operations: jitter regulation and link scheduling. This work is related to jitter regulation.

There are two main ways to quantify jitter [1]: one measure, called delay-jitter is the maximum difference between arrival times of different packets and the ideal time difference in a perfectly periodic sequence (where packets are spaced exactly $X_a$ time units apart, $X_a$ is the IAT of the packet arrival sequence). The second measure is rate-jitter; it bounds the difference in packet delivery rates at various times. More precisely, it measures the difference between the maximal and minimal IATs. Rate-jitter is a useful measure for many real-time applications such as voice and video broadcast over the Internet.

The rest of the paper is organized as follows: In Section II, we discuss the literature related to jitter control techniques in PSNs and queueing models with state dependent service. In Section III, we briefly discuss about MMPP, the 4-state MMPP model used in this work and two main assumptions about MMPP which makes the analysis of queueing model easier. Section IV discusses the proposed rate-jitter control algorithms and the analytical bound of rate-jitter for algorithm-A. In Section V, we discuss the $MMPP/\widetilde{D}/1/B_{on}$ queue modeling and we give analytical expression for the steady state queue length distribution. In Section VI, we give simulation results and finally, we conclude by summarizing the results and discussing future work in Section VII.

## II. Related Work

Mansour *et al*. [1] used *competitive analysis* in order to compare an on-line algorithm with off-line algorithm. An *off-line algorithm* schedules a packet by using future arrivals also. Though off-line algorithm is impractical, it does deliver departure/dequeueing sequence with minimum possible jitter and forms the lower bound. Hence, off-line algorithm is used to compare the performance of any on-line algorithm proposed (for the same packet arrival sequence). An *on-line algorithm* schedules a packet based on the packet arrival times on or before the service initiation instant of the packet in question. Mansour *et al*. proposed an on-line algorithm for rate-jitter control (we call it as Mansour algorithm in rest of the paper), which achieves a rate-jitter bounded by the rate-jitter of an off-line algorithm. Mansour algorithm requires a buffer size of $B_{on} = 2B + h$, where $B$ is the buffer size of the off-line algorithm and $h$ is a space parameter. Mansour algorithm tightly schedules the packets within the given bounds $I_{max}$ and $I_{min}$ and achieves a rate-jitter not more than $I_{max} - I_{min}$, where $I_{max}$ and $I_{min}$ are the maximum and minimum bounds on the inter-departure times (IDTs) of the off-line algorithm, respectively. ElBatt *et al*. [2] proposed a traffic recovery mechanism to control the jitter. A detailed survey of rate-control algorithms can be found in [3]. Hay *et al.* [4] extended [1] to multiple streams and derived tight lower bounds for jitter regulation, both in off-line and on-line cases. An analysis of delay jitter control algorithms can be found in [5].

A new queueing model $G/\widetilde{D}/1/K$ is proposed in [6] to analyze the performance of the proposed adaptive timing method with state dependent service rates. A packet voice multiplexer is modeled as an $M/\widetilde{D}/1/K$ queue in [7], where the least significant bits of a voice packet are dropped during congestion period of multiplexer to reduce the queueing delay. The service time of a packet is determined depending on the buffer occupancy.

The main characteristic of Internet traffic is that its parameters (packet IAT, data transmission rate, etc.) are Long Range Dependent (LRD), i.e., a non-zero correlation exists in long-term time-scales. MMPP is a widely used arrival process in communication networks for modeling traffic whose arrival rate varies with respect to time. It can capture the correlation between IATs in the arrival process. Andersen *et al.* [10] illustrated that the superposition of four two-state MMPP's are sufficient to model the second-order self-similar behavior over large time-scales. The authors also proposed a fitting procedure for matching second-order properties of counts to that of a second-order self-similar process. Muscariello *et al.* [16] proposed a new MMPP traffic model that accurately models the LRD Internet traffic over time-scales. Yoshihara *et al.* [17] also proposed a fitting method for self-similar traffic based on the superposition of two-state MMPP's that matches the variance function over several time-scales. Nogueira *et al.* [18] extended [17] to accurately produce the self-similar traffic. The authors proposed a new fitting procedure that matches the complete distribution (besides variance) at each time scale.

Fischer *et al.* [10] did a detailed survey on MMPP and presented all the results about MMPP and queues with MMPP as input. A 4-state MMPP model is developed in [11] to characterize the behavior of aggregate input traffic in an ATM multiplexer. The performance of an ATM multiplexer with MMPP (using the model in [11]) as input is studied in [12] by making two assumptions on MMPP. The authors modeled the buffer as an $MMPP/D/1/K$ queue and calculated the queue length distribution, mean waiting time and cell loss probabilities. Choi *et al.* [13] analyzed a queueing system $MMPP/G_1, G_2/1/B$ with queue length dependent service times and then applied the results to cell discarding scheme in ATM networks. The authors have defined two service times depending on whether the buffer level is above or below a threshold.

The performance metrics studied in this paper are rate-jitter and mean waiting time of a packet in the queue (jitter-buffer). Our contribution is two-fold: (a) proposed algorithm-A and algorithm-B (both uses the same buffer size as Mansour algorithm) for rate-jitter control, shown that both of them achieves less rate-jitter compared with Mansour algorithm (Mansour work [1] has a lot of practical implication, this is the main reason for comparing our proposed algorithms with Mansour algorithm) and (b) statistical modeling of jitter-buffer as $MMPP/\widetilde{D}/1/B_{on}$ queue for the performance analysis of proposed algorithms. In the literature, queues with MMPP input and state dependent service are not studied thoroughly. To the best of our knowledge, this queueing model is not considered so far. We have defined a service time corresponding to each level of buffer occupancy and state of the MMPP, so, all the service times are distinct. Steady state queue length distribution at departure epochs is calculated from the queueing model. Simulation results are in congruent with the derived analytical results. Even though the algorithms are proposed for TDMoIP application, we believe that they will work for any application in PSN.

## III. 4-State MMPP Model

The MMPP is a doubly stochastic Poisson process whose arrival rate varies according to an $M$-state irreducible continuous time Markov chain (CTMC) [10]. MMPP can be viewed as a super-position of '$M$' independent Poisson processes, where switching among the processes is governed by an $M$-state CTMC, i.e. when the MMPP is in state $i$, arrival process is Poisson with rate $\lambda_i$.

The analysis presented here is based on two assumptions, like in [12]: (i) MMPP changes its state at departure epochs and (ii) probability of two or more state changes between two successive departures is essentially zero (here state change means switching from one Poisson process to other). If the MMPP is in state $j$ after a departure, then the arrival rate is $\lambda_j$ until the next departure (i.e. until the next state change).

The states of a 4-state MMPP are labeled from 0 to 3, as shown in Fig. 1, taken from [11]. State transitions occur only between adjacent states with the rates specified in Fig. 1. The duration of state $j$ is exponentially distributed with parameter

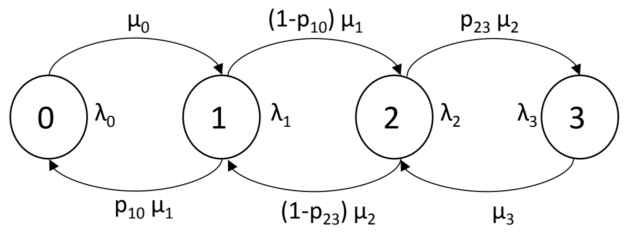

Figure 1: State transition diagram of a 4-state MMPP

$\mu_j$ and the arrivals in state $j$ follow Poisson process with parameter $\lambda_j$.

Let the MMPP has $M$ states, denoted by $j$ $(0 \le j \le M-1)$, and let $\lambda_j$ and $1/\mu_j$ be the arrival rate and mean state duration of state $j$ respectively and $S_{j,l}$ be the service time of a packet when the buffer level is $l$ and the MMPP is in state $j$. To satisfy the two assumptions, we need the mean time between arrivals and time between two successive departures (i.e. service time) should be much smaller than the mean state duration of the MMPP (this condition will be useful when the buffer is *non-empty*). Since the service times are multiple and distinct and number of states are $M$ (so, mean state durations are also multiple), for the service time to be smaller than mean state duration, maximum of the service times should be smaller than the minimum of mean state durations, i.e. $max\{S_{j,l}\}$ should be smaller than $min(1/\mu_j) = 1/\mu_{max}$, where $j \in \{0, 1, ..M-1\}$ and $l \in \{1, 2, ..B_{on}-1\}$. Similarly, $1/\lambda_j$ (mean time between arrivals), $\forall j$ should be smaller than $1/\mu_i$, $\forall i$ where $i, j \in \{0, 1, ..M-1\}$, i.e. $1/\lambda_{min}$ should be much smaller than $1/\mu_{max}$ (this condition will be useful when the buffer is *empty*).

## IV. Rate Jitter Control

For a given times sequence, the rate-jitter can be calculated from Equation (1) as follows:

$$Rate - jitter = \max_{0 \le i,j \le n} \{|(t_{i+1} - t_i) - (t_{j+1} - t_j)|\} \quad (1)$$

where $t_j$ is the arrival/departure instant of $j^{th}$ packet. When packets arrive at the destination, they are stored in the jitter-buffer and released some time later as discussed earlier. The release time (departure time) of a packet is determined by the rate-jitter control algorithm. Both the proposed algorithms starts with a buffer loading stage, the first packet is released only after the arrival of the $(B+1)^{th}$ packet and from there onwards packets are scheduled according to the algorithms.

### A. Assumptions

- Packets are of equal size, which is true in TDMoIP as mentioned previously.
- Buffer can hold integral number of packets.
- Packets arrive at destination in the order in which they are injected into PSN.
- Packets are processed in the FIFO discipline.

### B. Parameters, Notations and Definition

| | |
|---|---|
| $B$ | buffer size of an off-line algorithm |
| $1 \le h < B$ | space parameter for the on-line algorithm, such that $B_{on} = 2B + h$ |
| $I_{max}, I_{min}$ | maximum and minimum bounds on the IDT of an off-line algorithm |
| $X_a$ | average IAT in the input (and also the output) sequence and $I_{min} << X_a << I_{max}$ |
| $L_k$ | buffer level at the $k^{th}$ packet service initiation instant |
| $\widetilde{D}$ | set of $B_{on}-1$ state (queue length) dependent service times |
| $d_k$ | inter-departure time between $k^{th}$ packet and $(k-1)^{th}$ packet |
| $S(k)$ | service time of $k^{th}$ packet |

Define,

$$\delta_k \triangleq \left(\frac{B_{on} + 1 - L_k}{2B}\right) X_a \quad (2)$$

### C. Algorithm-A

This algorithm requires a buffer size of $B_{on}$, for each value of buffer occupancy $L_k$, it computes an IDT $d_k$, as given in the definition.

**Algorithm 1** Algorithm-A

```
1) Calculate $\delta_k$ using Equation (2)
2) if $0 \le L_k \le h$ then
     $d_k \leftarrow I_{max}$
3) else if $\delta_k > I_{min} + \frac{X_a}{B}$ and $L_k > h$ then
     $d_k \leftarrow \delta_k$
4) else if $\delta_k < I_{min} + \frac{X_a}{B}$ and $L_k > h$ then
     $d_k \leftarrow \delta_k + I_{min}$
5) end if
```

$X_a$, $I_{max}$ and $I_{min}$ are requirements for this algorithm. Assuming that $X_a$ is known in advance (like in ATM standard) is reasonable for real-time connections. Similarly $I_{max}$ and $I_{min}$ are the worst-case rate-jitter bounds, can be set based on the jitter bounds of the TDM traffic.

The following theorem calculates the rate-jitter bound of algorithm-A and proves that it is strictly less than that of [1].

*Theorem 1:* The rate-jitter of algorithm-A is bounded by $I_{max} - I_{min} - \frac{X_a}{B}$, which is strictly less than the rate-jitter bound $I_{max} - I_{min}$ of Mansour algorithm.

*Proof:* From the definition of $\delta_k$, we can observe that it is a discrete valued function, whose value decreases linearly (with slope $-\frac{X_a}{2B}$) with an increase in buffer level. The maximum and minimum values of $\delta_k$ are $\frac{B_{on}}{2B} X_a$ and $\frac{1}{2B} X_a$, they occur at $L_k = 1$ and $L_k = B_{on} - 1$ respectively.

We can observe from algorithm-A that the IDT in the output is $I_{max}$ when the number of packets $L_k$ are less than or equal to $h$ at the service initiation epoch of $k^{th}$ packet.

For any $L_k > h$, the IDT is less than $I_{max}$, this can be seen observed by substituting $L_k = h + 1$.
So, the maximum IDT in the output sequence is $I_{max}$.

Now consider $L_k = B_{on} - 1$, then
$\delta_k = \frac{2B+h+1-(2B+h-1)}{2B} X_a = \frac{X_a}{B} < I_{min} + \frac{X_a}{B}$
So, the IDT is $I_{min} + \frac{X_a}{B}$ from the definition.

For any $L_k < B_{on} - 1$, IDT is greater than $I_{min} + \frac{X_a}{B}$, so, this is the minimum IDT in the output sequence.

As mentioned previously, rate-jitter is calculated as the difference between maximum and minimum IDTs, as given in Equation (1). Therefore, the bound on the rate-jitter is, $I_{max} - (I_{min} + \frac{X_a}{B}) = I_{max} - I_{min} - \frac{X_a}{B}$, which is less than the rate-jitter of Mansour algorithm. ■

### D. Algorithm-B

This algorithm also requires a buffer size of $B_{on}$. The service time of a packet $k$ at the head of the buffer is a function of the number of packets $L_k$ at its service initiation instant. The service time $S(k)$ of $k^{th}$ packet is calculated as given in the definition.

**Algorithm 2** Algorithm-B

```
1) Calculate $\delta_k$ using Equation (2)
2) if $\delta_k > I_{min} + \frac{X_a}{B}$ and $L_k \leq B - h$ then
       $S(k) \leftarrow \delta_k + \frac{hX_a}{B}$
3) else if $\delta_k > I_{min} + \frac{X_a}{B}$ and $L_k > B - h$ then
       $S(k) \leftarrow \delta_k$
4) else if $\delta_k < I_{min} + \frac{X_a}{B}$ and $L_k > B - h$ then
       $S(k) \leftarrow \delta_k + I_{min}$
5) end if
```

The service times of algorithm-B (or IDTs of algorithm-A) are state (queue length) dependent. If the number of packets ($L_k$) in the queue increases, $\delta_k$ increases, so, the the service time (or IDT) decreases i.e. service rate increases. Service time (or IDT) is maximum when there are fewer number of packets in the buffer and minimum when the buffer is full/nearly full.

## V. $MMPP/\widetilde{D}/1/B_{on}$ QUEUE MODEL

We now model the jitter buffer with limited capacity $B_{on}$ as a queueing model with MMPP input and queue length dependent service times. When a packet departs from the queue, queue length can take any one of the values from 0 to $B_{on} - 1$. If the length of the queue is zero, then the algorithm has to wait for the next arrival and start serving it. So, buffer level $L_k$ cannot take zero while calculating service times. If the length of the queue is non-zero, the service time for the next packet is calculated from the queue length and present state of the MMPP at that instant. So, buffer level $L_k$ at the $k^{th}$ packet service initiation epoch takes any one of the values from 1 to $B_{on} - 1$.

Let us re-define,

$$\delta_{k,j} = \left( \frac{B_{on} + 1 - L_k}{2B} \right) X_{M_j} \tag{3}$$

where $X_{M_j}$ is the average inter-arrival time when the MMPP is in state $j$, so, $X_{M_j} = 1/\lambda_j$.
At the service initiation instant of $k^{th}$ packet, $\delta_{k,j}$ is calculated from Equation (3). Depending on the value of $\delta_{k,j}$, the service time of $k^{th}$ packet is calculated according to the algorithm we use. So, $(B_{on} - 1)$ deterministic service times are possible for each state of the MMPP. Therefore, $M(B_{on} - 1)$ service times: $\{S_{j,l} \; : \; 0 \leq j \leq M - 1, 1 \leq l \leq B_{on} - 1\}$ are possible. This allows us to model the jitter buffer as an $MMPP/\widetilde{D}/1/B_{on}$ queue, where $\widetilde{D}$ represents a set of $M(B_{on} - 1)$ state dependent service times.

We observe the state of the MMPP and the number of packets in the queue at departure epochs. Let $J_k$ and $L_k$ be the state of the MMPP and the number of packets in the queue at (i) $(k-1)^{th}$ packet departure or (ii) $k^{th}$ packet service initiation instant, both are same except if the $(k-1)^{th}$ packet leaves the system empty. Let $L_{k+1}$ be the same for $(k+1)^{th}$ packet and let $A_k$ be the number of arrivals during the service time of $k^{th}$ packet. Then the following recursive relation holds good:

$$\hat{L_{k+1}} = \begin{cases} L_k + A_k - 1, & if\, L_k > 0 \\ A_k\,, & if\, L_k = 0 \end{cases} \tag{4}$$

Now,

$$L_{k+1} = min(B_{on} - 1,\, \hat{L_{k+1}}) \tag{5}$$

From Equations (4) and (5), it is clear that $L_{k+1}$ depends only on $L_k$ and $A_k$. Since arrivals in each state of the MMPP follows Poisson process, $A_k$ is i.i.d and the state transitions of the MMPP are governed by a CTMC as mentioned previously. Therefore, $\{J_k, L_k,\, k \geq 0\}$ forms an embedded Markov chain (EMC) with the finite state space $\{0, 1, 2, ..., M-1\}X\{0, 1, 2, ..., B_{on} - 1\}$. At any arbitrary time instant the state of the system is represented by the pair $(j, l)$. The 1-D representation of this state space is $\{(0,0), (0,1), ..., (0, B_{on} - 1), (1,0), (1,1), ..., (1, B_{on} - 1), .....(M-1, 0), (M-1, 1), ..., (M-1, B_{on}-1)\}$. So, we can model the state $(j, l)$ as an an EMC, the transition probability matrix $P$ of the EMC is given in Equation (6),

$$P = [P_{ij}] = P(present\, state = i,\, next\, state = j)$$

$$P = \begin{bmatrix} P_{0,0} & P_{0,1} & \cdots & \cdots & P_{0,M-1} \\ P_{1,0} & P_{1,1} & \cdots & \cdots & P_{1,M-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ P_{M-1,0} & P_{M-1,1} & \cdots & \cdots & P_{M-1,M-1} \end{bmatrix} \tag{6}$$

where the elements $P_{ij}$ are sub-matrices of size $B_{on}XB_{on}$ [12] and $P$ is of size $MXM$.

Since we have assumed that multiple state transitions cannot occur between two successive departures, $P_{i,j} = 0$, for $|i - j| > 1$. So, $P$ matrix will have elements in main diagonal and in its two co-diagonals (one above and one below the main diagonal). Elements of the sub-matrix $P_{i,j}$ are denoted as $P_{i,j}(m, l)$ given by,

$$P_{i,j}(m, l) = P(J_{k+1} = j,\, L_{k+1} = l \mid J_k = i,\, L_k = m) \tag{7}$$

which is the probability of $l$ packets in the queue when the MMPP is in state $j$ after a departure given that there were $m$

$$P_{i,j} = \begin{bmatrix} \alpha(0;S_{j,1},\lambda_j)\beta(i,j;S_{j,1}) & \alpha(1;S_{j,1},\lambda_j)\beta(i,j;S_{j,1}) & \alpha(2;S_{j,1},\lambda_j)\beta(i,j;S_{j,1}) & \cdots & \alpha(B_{on}-2;S_{j,1},\lambda_j)\beta(i,j;S_{j,1}) & 1-\sum_{k=0}^{B_{on}-2}\alpha(k;S_{j,1},\lambda_j)\beta(i,j;S_{j,1}) \\ \alpha(0;S_{j,1},\lambda_j)\beta(i,j;S_{j,1}) & \alpha(1;S_{j,1},\lambda_j)\beta(i,j;S_{j,1}) & \alpha(2;S_{j,1},\lambda_j)\beta(i,j;S_{j,1}) & \cdots & \alpha(B_{on}-2;S_{j,1},\lambda_j)\beta(i,j;S_{j,1}) & 1-\sum_{k=0}^{B_{on}-2}\alpha(k;S_{j,1},\lambda_j)\beta(i,j;S_{j,1}) \\ 0 & \alpha(0;S_{j,2},\lambda_j)\beta(i,j;S_{j,2}) & \alpha(1;S_{j,2},\lambda_j)\beta(i,j;S_{j,2}) & \cdots & \alpha(B_{on}-3;S_{j,2},\lambda_j)\beta(i,j;S_{j,2}) & 1-\sum_{k=1}^{B_{on}-2}\alpha(k;S_{j,2},\lambda_j)\beta(i,j;S_{j,2}) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & \alpha(0;S_{j,B_{on}-1},\lambda_j)\beta(i,j;S_{j,B_{on}-1}) & 1-\alpha(0;S_{j,B_{on}-1},\lambda_j)\beta(i,j;S_{j,B_{on}-1}) \end{bmatrix} \quad (8)$$

packets in the queue and the MMPP was in state $i$ after the previous departure.

Because of the assumptions we made, $P_{i,j}(m,l)$ can be calculated as the product of $P(L_{k+1} = l \mid J_k = i, L_k = m;$ in the service time $S_{j,l})$ and $P(J_{k+1} = j \mid J_k = i;$ in the service time $S_{j,l})$, where $S_{j,l}$ is the service time of a packet when the buffer level is $l$ and MMPP is in state $j$. The sub-matrix $P_{i,j}$ is given in Equation (8), where $\alpha(k; S_{j,l}, \lambda_j)$ is given below:

$$\alpha(k; S_{j,l}, \lambda_j) = \frac{e^{-\lambda_j S_{j,l}}(\lambda_j S_{j,l})^k}{k\,!},\ k \geq 0 \qquad (9)$$

which is the probability of $k$ arrivals in the service time $S_{j,l}$ when the arrival rate is $\lambda_j$; and $\beta(i, j; S_{j,l})$ is the probability of the MMPP to change its state from $i$ to $j$ in the service time $S_{j,l}$. If MMPP changes its state from $i$ to $j$, this is equivalent to $|(i-j)|$ arrivals (equal to one arrival, according to the assumptions) in the service time $S_{j,l}$. As mentioned previously, the duration of state $j$ is exponentially distributed with parameter $\mu_j$, so, we can write $\beta(i, j; S_{j,l})$ as follows:

$$\beta(i,j;S_{j,l}) = \begin{cases} \text{P(MMPP will not change its state in service time } S_{j,l}), & \text{if } i = j \\ \text{P(MMPP will change its state from i to j in service time } S_{j,l}), & \text{otherwise} \end{cases}$$

$$= \begin{cases} \frac{e^{-\mu_j S_{j,l}}(\mu_j S_{j,l})^0}{0\,!}, & \text{if } i = j \\ \frac{e^{-\lambda_j S_{j,l}}(\lambda_j S_{j,l})^1}{1\,!}, & \text{otherwise} \end{cases}$$

Let $\pi = \{\pi_{j,l} : 0 \leq j \leq M-1, 0 \leq l \leq B_{on}-1\}$ be steady state probability vector of the states at departure epochs, where $\pi_{j,l}$ denotes the steady state probability of the state $(j,l)$. By solving $\pi = \pi P$ [8], we get the steady state probability vector of state. Now, the steady state probability of buffer occupancy $(\pi_l,\ 0 \leq l \leq B_{on}-1)$ is calculated as,

$$\pi_l = \sum_{j=0}^{M-1} \pi_{j,l}. \qquad (10)$$

Now, from [12], we can calculate queue length distribution at arrival epochs, from which mean waiting time and packet loss probability calculation is straightforward.

## VI. Simulation Results and Discussion

Simulation parameters are taken in such a way that the two assumptions gets satisfied always. We have taken, the average arrival rates in each state of the MMPP as $\lambda_0 = 0.6\,pptu$ (***packets per time unit***), $\lambda_1 = 0.7\,pptu$, $\lambda_2 = 0.8\,pptu$, $\lambda_3 = 0.9\,pptu$, where one time unit depends on the line speed and packet size. Mean state durations are exponentially distributed and are set as $\mu_0 = 0.02$, $\mu_1 = 0.05$, $\mu_2 = 0.015$, $\mu_3 = 0.001$ and, $X_a$ is taken as $1/mean(\lambda_0, \lambda_1, \lambda_2, \lambda_3)$ in the simulation. In Fig. 2, buffer size $B$ varied from 4 to 30, and set $h = 2$, so, $B_{on}$ ranges from 10 to 62.

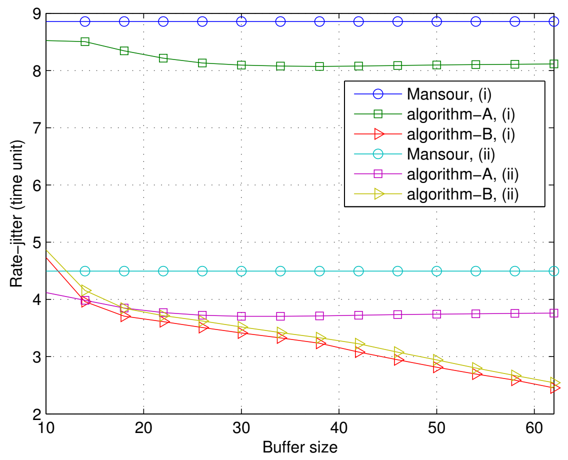

Figure 2: Comparison of rate-jitter for different algorithms in both cases

*Choice of* $I_{max}$ *and* $I_{min}$: One trivial choice for $I_{max}$ and $I_{min}$ is $X_{max}$ and $X_{min}$ [1], where $X_{max}$ and $X_{min}$ are the maximum and minimum IATs in the input sequence, respectively. But by using tighter $I_{max}$ and $I_{min}$, we may get a stronger rate-jitter guarantee, i.e., we may achieve an $I_{max}$ less than $X_{max}$ and an $I_{min}$ greater than $X_{min}$ using off-line algorithm. It is not possible to give an exact lower bound of $I_{max}$, since we do not have control over input arrival process. In this paper, we have simulated two **cases**: **(i)** $I_{max} = X_{max}$, $I_{min} = X_{min}$ and **(ii)** $I_{max} = 0.5X_{max}$, $I_{min} = 2X_{min}$. When we reduce $I_{max}$ value further beyond $0.5X_{max}$, we have arrived at a situation where a packet is scheduled before its arrival. The reason for this is the maximum IDT $I_{max}$ is smaller than the IATs of packets arrived in that situation.

From Fig. 2, it is evident that the rate-jitter of algorithms A and B is less compared with Mansour algorithm in both cases. Among the three algorithms, algorithm-B achieves less rate-jitter. Since $I_{max}$ in case (i) is larger than that of case (ii), rate-jitter is more in case (i) compared with case (ii). For one realization of input, the maximum and minimum IDTs

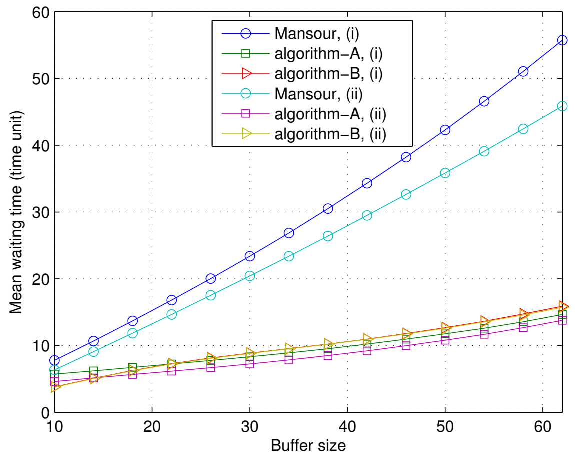

Figure 3: Comparison of mean waiting time of a packet for different algorithms in both cases

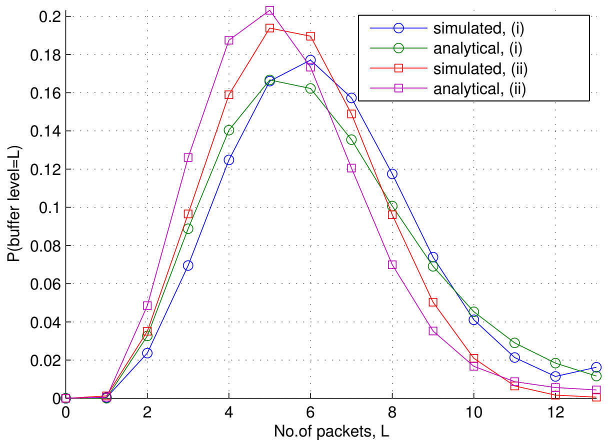

Figure 4: Queue length distribution analytical vs. simulation for both cases when algorithm-A is applied

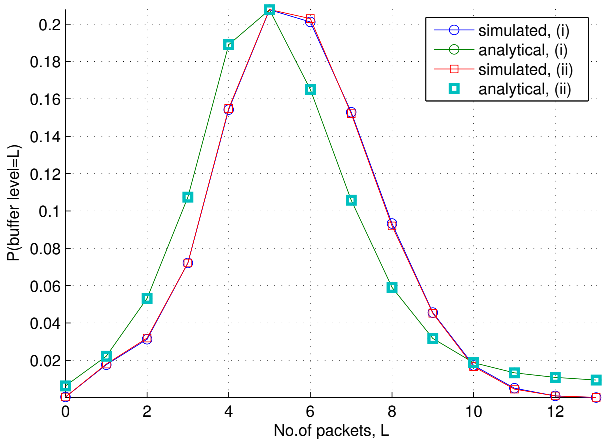

Figure 5: Queue length distribution analytical vs. simulation for both cases when algorithm-B is applied

in the output sequence of Mansour algorithm are $I_{max}$ and $I_{min}$, respectively, for all buffer sizes. Hence, the rate-jitter is constant.
The maximum IDT is constant in algorithm-A for all buffer sizes, but minimum IDT decreases with an increase in buffer size B up-to some value and from then onwards it is almost constant. Because at fewer values of buffer size, it becomes full and the minimum IDT takes the minimum of the possible values it can take, but at high buffer values it may not happen. Therefore, rate-jitter for algorithm-A decreases up-to some buffer size and almost constant from there in both cases. For algorithm-B, both the maximum and minimum IDTs decreases with an increase in the buffer size. Therefore, rater-jitter keep on decreasing for increasing buffer size in both cases.

The mean waiting time of a packet for different algorithms is calculated from simulation. From Fig. 3, it is observed that in Mansour algorithm, the waiting time increases like an exponential as a function of buffer size B, which is large compared with algorithms A and B in both cases. Mean waiting time of algorithm-A (algorithm-B) is almost equal in both cases.

To simulate the queue length distribution, we have taken $B = 6$, $h = 2$, so, $B_{on} = 14$ for both cases. For algorithm-A, IDTs are taken as service times to find the queue length distribution, this is true as long as the buffer is not empty. Buffer becomes empty rarely because of the following: (a) when the buffer level decreases, algorithm-A schedules the packets in such a way that the IDT is $I_{max}$, so, the probability of an arrival before the buffer becomes empty is high. (b) since arrivals follow Poisson process in each state of the MMPP, there is a high probability for the IATs (follows Exponential distribution) to take small values. So, most of the time there will be a packet in the buffer. This is true from TDMoIP perspective also because the incoming rate has to be equal to the outgoing rate (utilization, $\rho = 1$).

From Fig. 4 and Fig. 5, we can observe that the analytical queue length distribution curves for algorithm-A and algorithm-B, respectively, match with the simulation results for both cases. Since algorithm-B doesn't depend on $I_{max}$, the queue length distribution curves (analytical or simulated) are almost same irrespective of the case, which can be observed from Fig. 5. Even though algorithm-B depends on $I_{min}$, its value is so small that cannot influence the curves.

## VII. Conclusions and Future Work

In this paper, we have proposed two rate-jitter control algorithms for TDMoIP application. For algorithm-A, the bound on rate-jitter is calculated analytically and is shown that is smaller as compared with Mansour algorithm. The simulation results show that the performance of algorithm-A and algorithm-B is better than Mansour algorithm with respect to mean waiting time and rate-jitter. Therefore, algorithm-B is superior to both algorithm-A and Mansour algorithm, but the analytical bound of algorithm-B is yet to be proved. We have modeled the jitter-buffer as an $MMPP/\widetilde{D}/1/B_{on}$ queue, derived an expression for the queue length distribution. The simulation results for both algorithms match the analytical queue length distribution.

The future work is directed towards the study of performance of proposed algorithms in multiple streams environment. The statistical analysis of algorithm-B is under study. We also aim to calculate an analytical expression for variance of the departure process of the proposed queueing model. As a future work, we are extending this work with different self-similar processes as input, because the real-time traffic in IP can be best modeled using these arrival processes. We also aim to determine the waiting time distribution.

## References

[1] Y. Mansour and B. Patt-Shamir, “Jitter control in QoS networks,” IEEE/ACM Trans. on Networking, Vol.9, No.4, Aug 2001, pp. 492-502.

[2] T. A. ElBatt, S. El-Henaoui, and S. Shaheen, “Jitter recovery strategies for multimedia traffic in ATM networks,” in Proc. GLOBECOM’96, vol.2, 1996, pp. 1202-1206.

[3] H. Zhang and S. Keshav, “Comparison of rate-based services disciplines,” in Proc. ACM SIGCOMM, 1991, pp. 113-121.

[4] D. Hay and G. Scalosub, “Jitter regulation for multiple streams,” ACM Trans. on Algorithms, Vol.6, No.1, Article 12, Dec 2009, pp. 12.1-12.19, doi: 10.1145/1644015.1644027.

[5] S. Jagadish and R. Manivasakan, “Analysis of jitter control algorithms in QoS networks,” in Proc. Second Asian Himalayas International Conference on Internet (AH-ICI), Nov 2011.

[6] Z. Yan and J. Yih-Chyun, “Performance analysis of adaptive timing method for voice over ATM using queuing models,“ in Proc. of Winter International Symposium on Information and Communication Technologies (WISICT’04), 2004, pp. 1-8.

[7] K. Sriram and D. M. Lucatoni, ”Traffic smoothing effects of bit dropping in a packet voice multiplexer,“ IEEE Trans. on Comm.,Vol.37, No.7, July 1989, pp. 703-712.

[8] U. Narayan Bhat, ”An Introduction to queueing theory: Modeling and analysis in applications,“ Springer, 2008.

[9] D. Gross and C. M. Harris, “Fundamentals of queueing theory,” New York: Wiley, 1985, 2nd ed., pp. 279-285.

[10] W. Fischer and K. Meier-Hellstern, “The Markov-modulated Poisson process (MMPP) cookbook,” in Perf. Eval, Elsevier, Vol.18, No.2, Sep 1993, pp. 149-171.

[11] F. Yegenoglu and B. Jabbari, “Modeling of aggregated bursty traffic sources in ATM multiplexers,” in Proc. ICC’93, May 1993, pp. 1703-1707.

[12] F. Yegenoglu and B. Jabbari, “Performance evaluation of MMPP/D/1/K queues for aggregate ATM traffic models,” in Proc. INFOCOM’93, 1993, pp. 1314-1319.

[13] B. D. Choi and D. I. Choi, “Queueing system with queue length dependent service times and its application to cell discarding scheme in ATM networks,” in Proc. IEE, 1996, pp. 5-11.

[14] Madhu Sikha and R. Manivasakan, “Novel Rate-Jitter Control Algorithms for TDMoIP, ” in IEEE National Conference on Communications (NCC), 2013, pp. 1-5.

[15] A. T. Andersen and B. F. Nielsen, “A Markovian approach for modeling packet traffic with long-range dependence,” IEEE Journal on Selected Areas in Comm., Vol.16, No.5, June 1998, pp. 719-732.

[16] L. Muscariello, M. Mellia, M. Meo, M. Ajmone Marsan, and R. Lo Cigno, “Markov models of internet traffic and a new hierarchical MMPP model,” Journal on Computer Communications, Elsevier, Vol.28, No.16, Oct 2005, pp. 1835-1851.

[17] T. Yoshihara, S. Kasahara, and T. Takahashi, “Practical time-scale fitting of self-similar traffic with Markov-modulated Poisson process,” Telecommunication Systems, Vol. 17, No. 1-2, 2001, pp. 185-211.

[18] A. Nogueira, P. Salvador, R. Valadas, and A. Pacheco, “Modeling self-similar traffic through Markov modulated Poisson processes over multiple time scales,” in Proc. 6th IEEE International Conference on High Speed Networks and Multimedia Communications (HSNMC’03), July 2003.

[19] Y. Stein, R. Shashoua, R. Insler, and M. Anavi,“Time Division Multiplexing over IP (TDMoIP),” RFC 5087, Dec 2007.

[20] P. M. Fernandez, “Circuit Switching in the Internet’,” Doctoral thesis, Chapter 2, Stanford University, June 2003.

[21] Y. Stein and E. Schwartz, “Circuit Extension over IP: An Evolutionary Approach to Transporting Voice and Legacy Data over IP Networks,” RAD Data Comm., March 2001.